Shell-model CI codes and applications

Calvin Johnson(1)
Plamen Krastev(1,2)*
Erich Ormand (2)

[1]San Diego State University
[2]Lawrence Livermore National Lab
* = supported by UNEDF

Summary:
* Major rewrite of REDSTICK CI shell-model code
* Significant improvement in performance and potential
* Hosted mini-workshop on Leadership-class CI codes at SDSU
* Physics applications currently underway
-> two applications:
  * study of errors arising from model space truncation
  * exact vs. Fermi gas calculation of level densities

UNEDF Pack Forest Meeting June 2009

Shell-model CI codes and applications

Past year's work on REDSTICK:

July 2009: P. Krastev joined effort
August 2009: Analysis of performance of REDSTICK and modeling parallelization algorithms
-> need to reorganize representation of "jumps"

Krastev: **modeled parallel distribution** of jumps
Johnson & Ormand: major rewrite of code

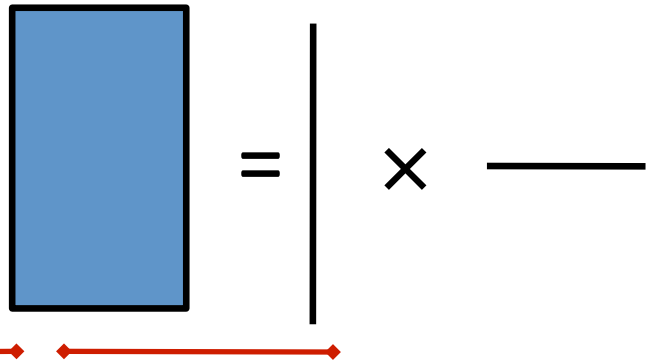
Feb 2009: revised 2-body version of code finished
-- **set-up is 10x faster** than old version
March 2009: further improvements:
-- Hamiltonian application is at least **2x faster** than old version (fixed bottleneck from old version)
-- began implementation of **new parallelization scheme**
April 2009: Implemented **thick-restart Lanczos**

Shell-model CI codes and applications

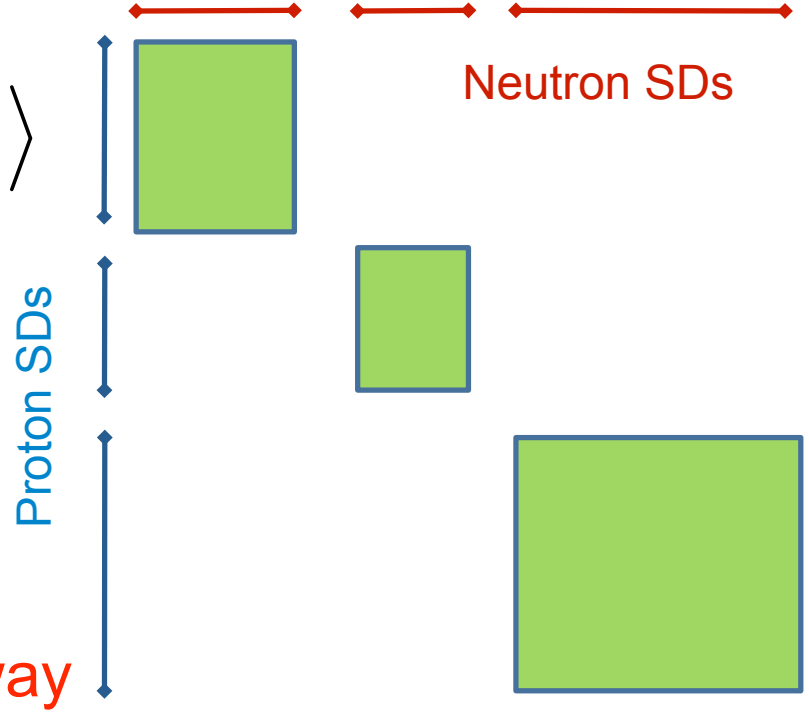
Key idea for on-the-fly algorithms for the shell model

Represent an area by its boundary
→ **Factorization** of problem

→ **Reduces memory load**

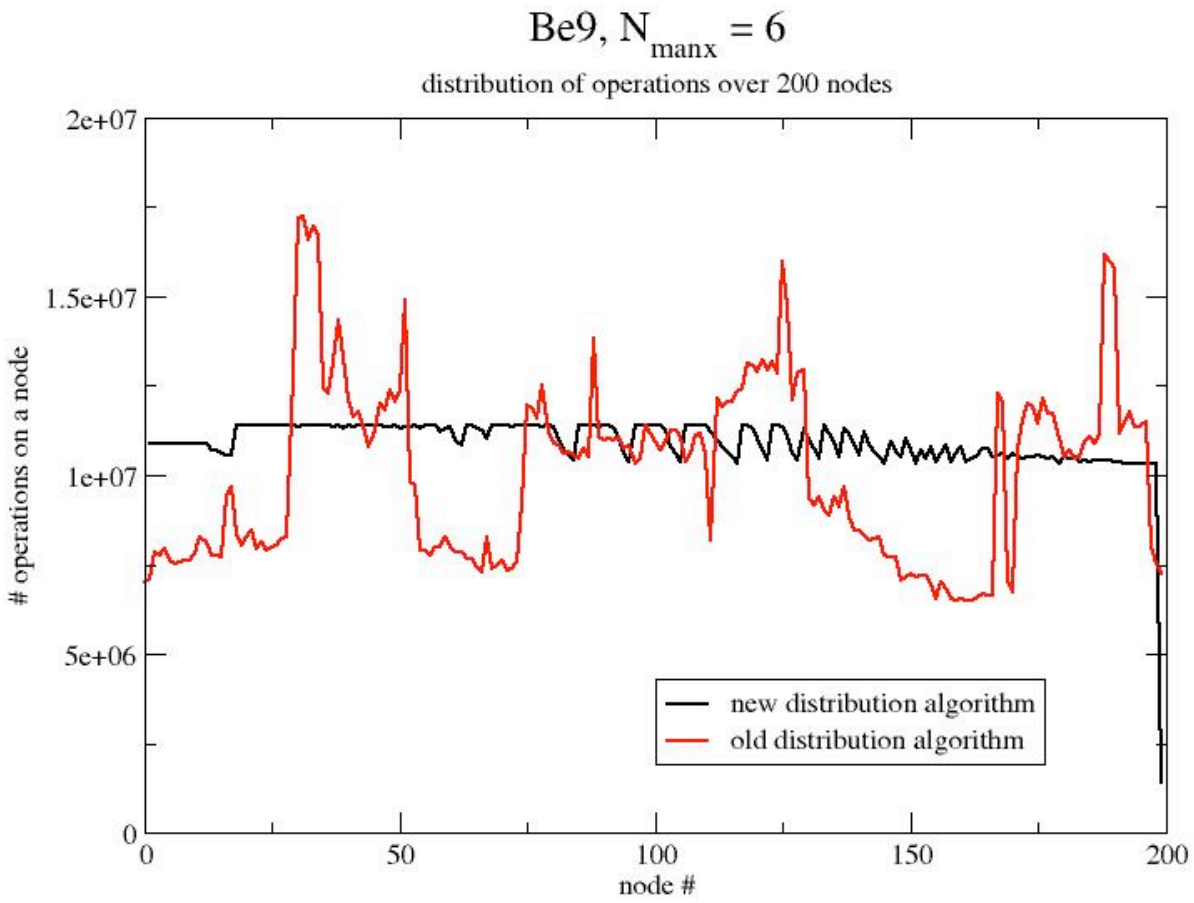


$$|\alpha\rangle = |\alpha_p\rangle \times |\alpha_n\rangle$$



Hamiltonian can be
factorized in same way

Shell-model CI codes and applications



REDSTICK ➔ BIGSTICK

UNEDF Pack Forest Meeting June 2009

Shell-model CI codes and applications

Why "on-the-fly"?

Factorization of Hamiltonian -> reduced memory
-> larger problem on same machine

Comparison of RAM requirements (2-body interactions only)

Does not include lanczos vector storage

Nuclide	Space	Basis dim	matrix store	on-the-fly
^{56}Fe	*pf*	501 M	290 Gb	0.72 Gb
^{7}Li	N_{max}=12	252 M	3600 Gb	96 Gb
^{7}Li	N_{max}=14	1200 M	23 Tb	624 Gb
^{12}C	N_{max}=6	32M	196 Gb	3.3 Gb
^{12}C	N_{max}=8	590M	5000 Gb	65 Gb
^{12}C	N_{max}=10	7800M	111 Tb	1.4 Tb
^{16}O	N_{max}=6	26 M	142 Gb	3.0 Gb
^{16}O	N_{max}=8	990 M	9700 Gb	130 Gb

Shell-model CI codes and applications

Comparison of RAM requirements (3-body interactions) - Estimate

Nuclide	Space	Basis dim	store	on the fly
^{7}Li	N_{max}=12	252 M	100 Tb	2.6 Tb
^{7}Li	N_{max}=14	1200 M	760 Tb	20 Tb
^{12}C	N_{max}=6	32M	4 Tb	0.07 Tb
^{12}C	N_{max}=8	590M	180 Tb	3 Tb
^{12}C	N_{max}=10	7800M	5000 Tb	86 Tb

1 Tb requires approximately 1000 cores (depending on architecture)

Shell-model CI codes and applications

In March 2009 we hosted a mini-workshop on
Leadership Class CI codes at SDSU

Attendees:
Johnson, Krastev, Ormand
Vary, Maris
Navratil
Horoi
Ng, Yang

An excellent chance to "look under the hood" of our algorithms and share ideas for next-generation calculations;
these discussions have led to mutual improvements in our CI codes → cooperation not competition!

Shell-model CI codes and applications

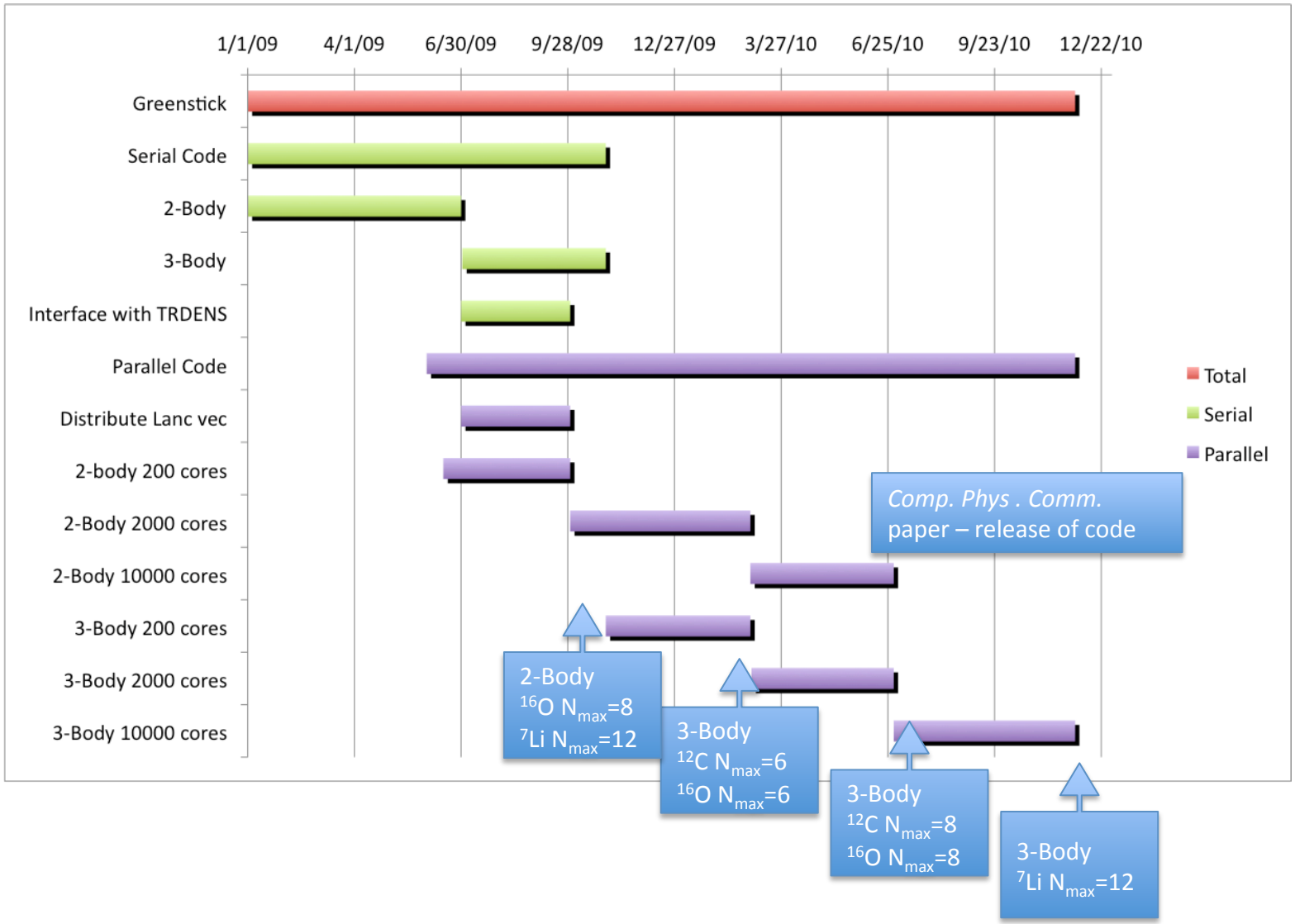
Preliminary physics applications (serial so far):

* Comparison of exact results with pairing approximations in *pf* shell (with N. Sandulescu; ran full ^{52}Fe in 30 hrs)
* Calculation of *sd* shell nuclei in 2hw space to study effects of model space truncation (with G. Bertsch; ran many cases with 20-50M states, approx 1-2 days each)
* EFT in atomic gases (with I. Stetcu et al; 4 particles up to 16hw)

Applications this fall:

* Electric polarizability in A = 6 (with I. Stetcu et al)
* *p*-shell nuclides (P. Navratil)
* Isospin breaking and CKM unitarity in *pf*-shell nuclei (EO+CJ)

Shell-model CI codes and applications



UNEDF Pack Forest Meeting June 2009

Shell-model CI codes and applications

New application: effects of shell model space truncation (with George Bertsch)

Motivation: Shell-model calculations work in finite many-body space, using either phenomenological fits (a la Alex Brown) or renormalization scheme. Such truncations induce many-body interactions which may not be part of the fit—thus the rms error has an intrinsic limitation.

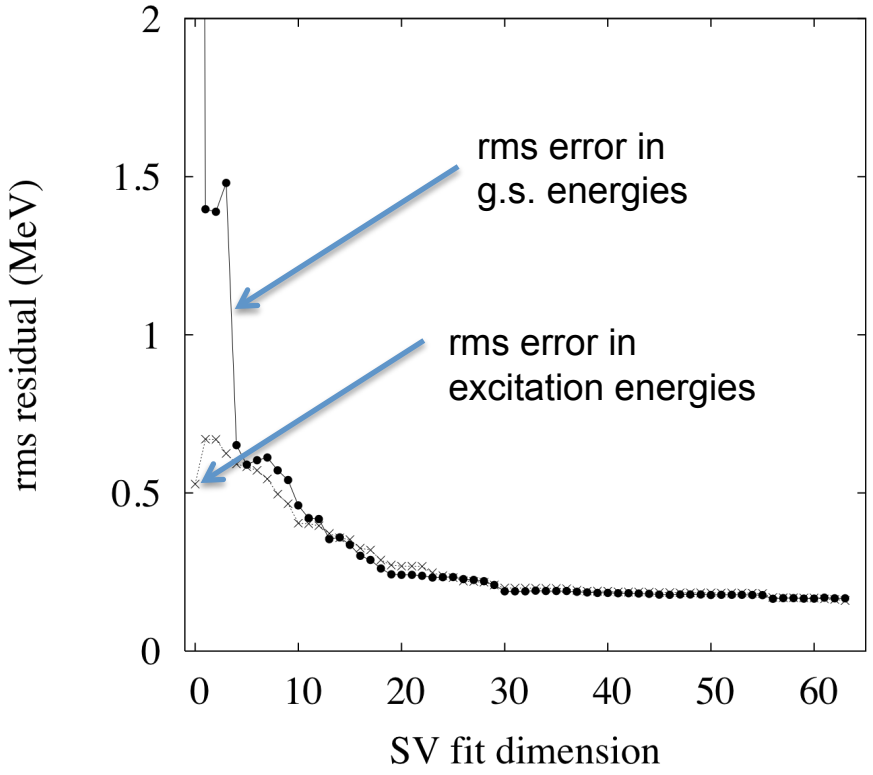
Methodology: Generate "exact" spectrum in large model space (*sd* + 2p-2h *pf* + 1p-1h *sdg*); adjust USD interaction in *sd* shell in least-squares fit of "exact" spectrum. **How much can we reduce the rms error?**

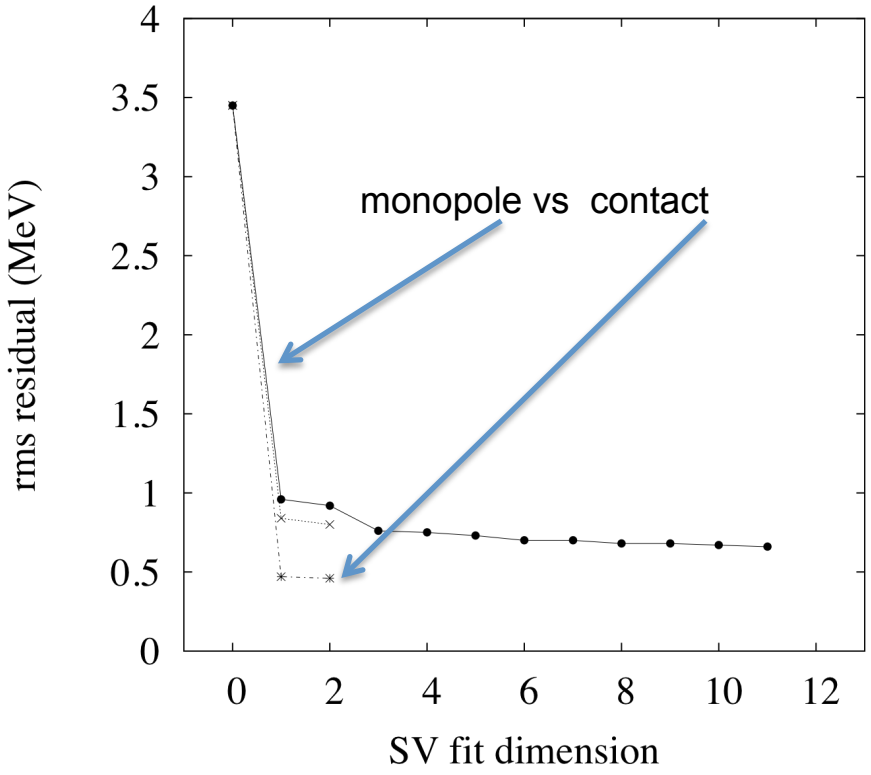
Shell-model CI codes and applications

New application: effects of shell model space truncation

$$\hat{H} \rightarrow \hat{H} + \Delta\hat{H}$$

Do least-squares fit using SVD





Shell-model CI codes and applications

New application: effects of shell model space truncation

Lessons learned:

As expected, one cannot exactly reproduce the "full space" calculation using only two-body interactions in the smaller space (many-body forces induced by truncation)

Least-squared fit dominated by a handful of interactions

Two "simple" forces do this most of the renormalization:
either
* isoscalar contact interaction (best); or
* N(N+1) ("monopole")

Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density

Motivation: Many approaches to nuclear level density begin with Bethe's Fermi gas model, usually with strong, phenomenological corrections. How good/bad is it?

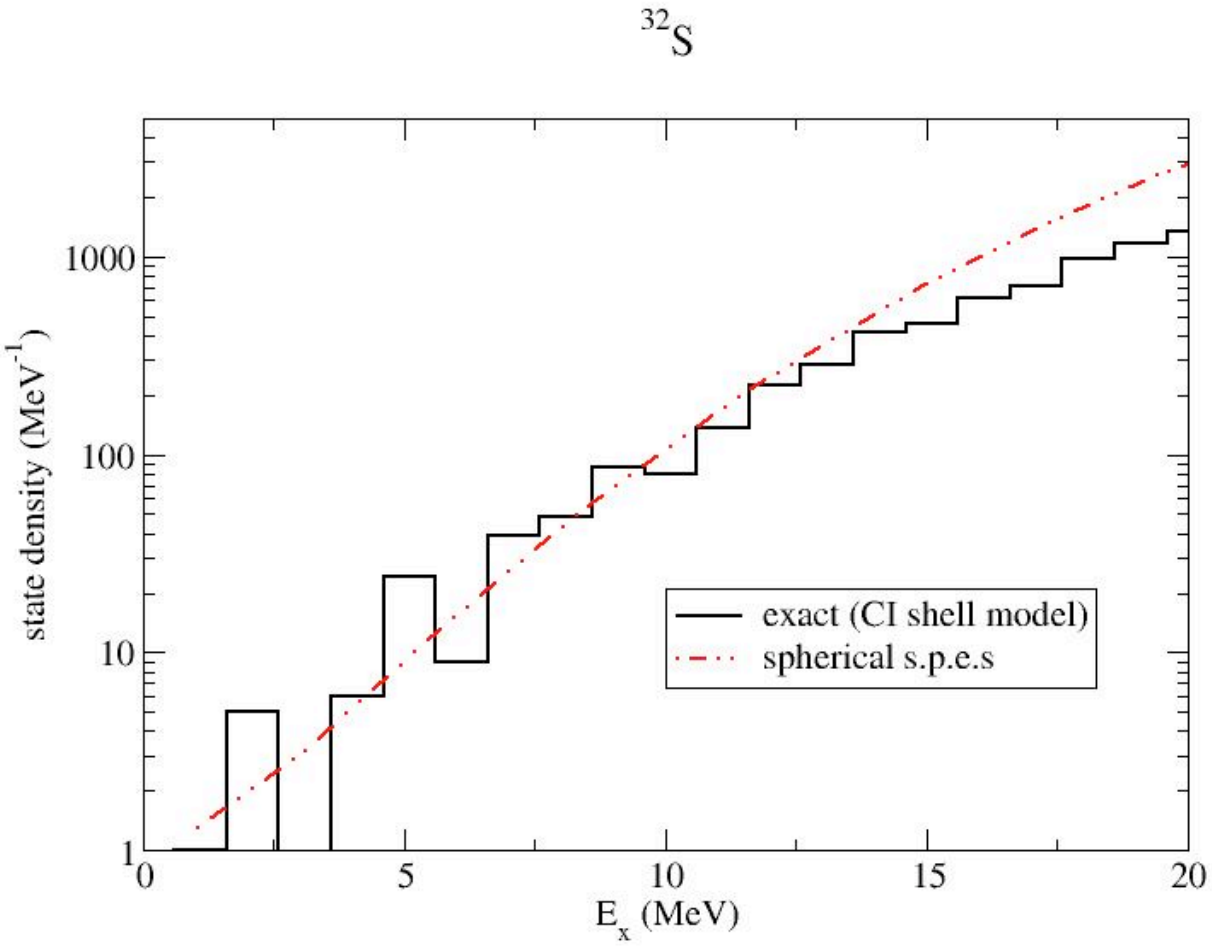
Methodology: Compute exact level (state) density with CI code. Using same model space & input Hamiltonian, compute HF s.p.e.s and compute Fermi gas partition function:

$$\ln Z(\alpha,\beta) = \sum_i \ln\left(1+\exp(\alpha-\beta\varepsilon_i)\right)$$

Invert Laplace transform to get density of states

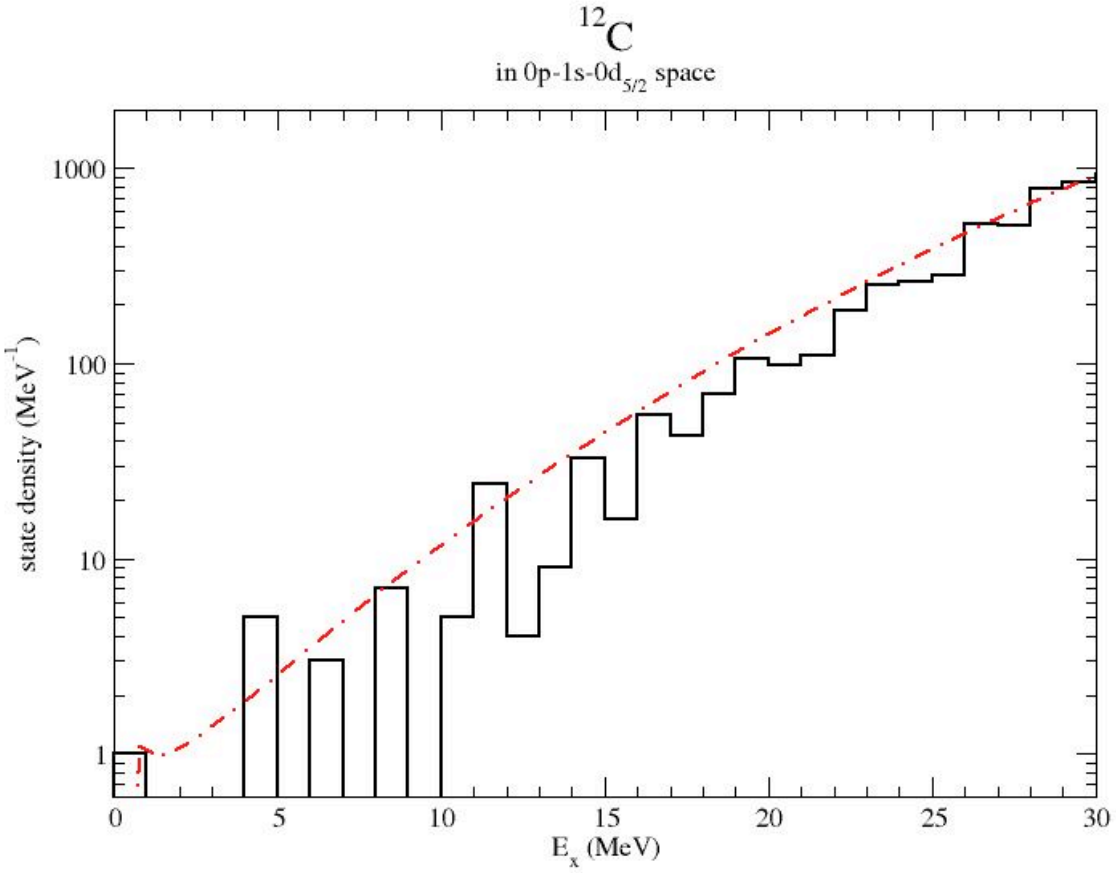
Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density



Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density



Shell-model CI codes and applications

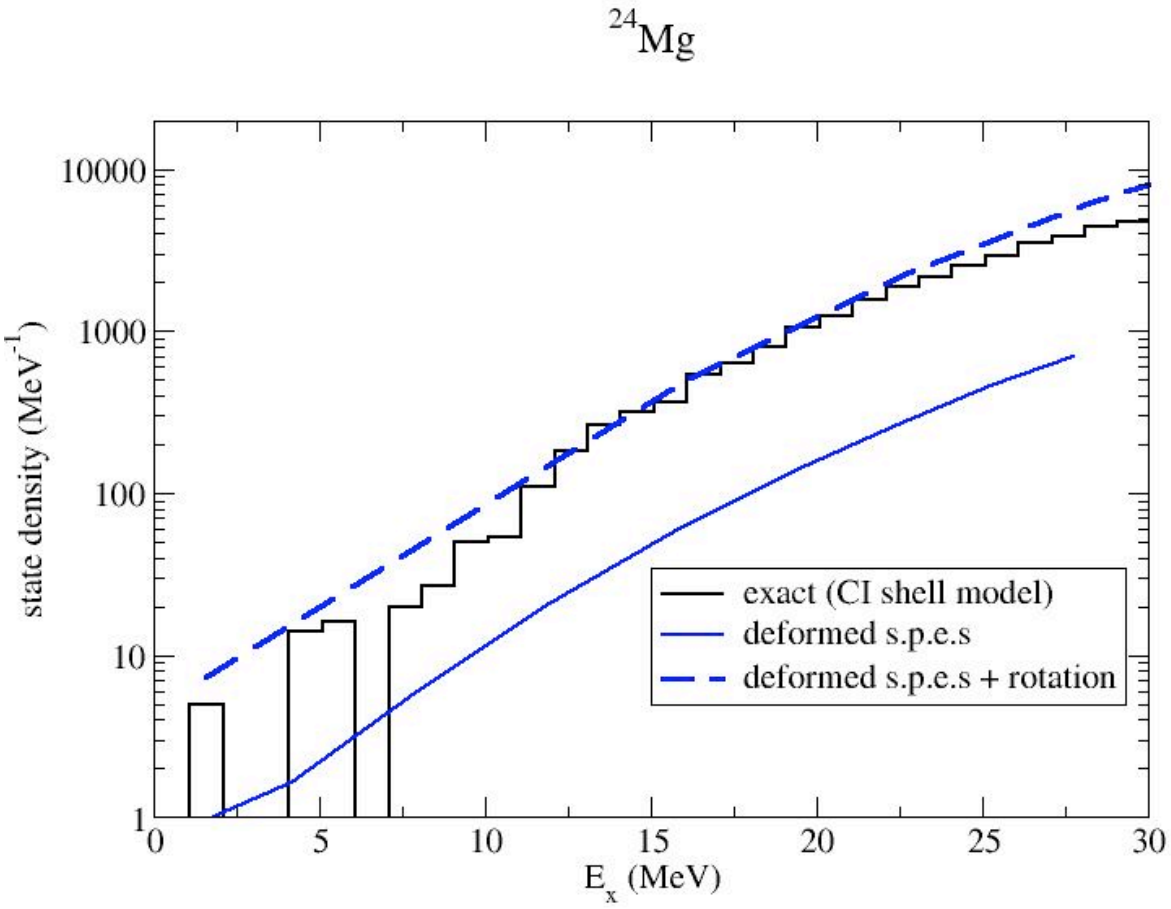
New application: tests of a cheap method to compute the level density

What if nuclei aren't spherical?
Add rotational partition function *with parameters from cranked HF*

$$Z_{rot} = \sum_J (2J+1) a_J^2 \exp(-\beta E_J) \approx \frac{\sqrt{\pi}\bar{J}}{\left(1+\beta \bar{E}_{rot}\right)^{3/2}}, \bar{J} = \sqrt{\langle J(J+1)\rangle}, \bar{E}_{rot} = \frac{\bar{J}^2}{2I}$$

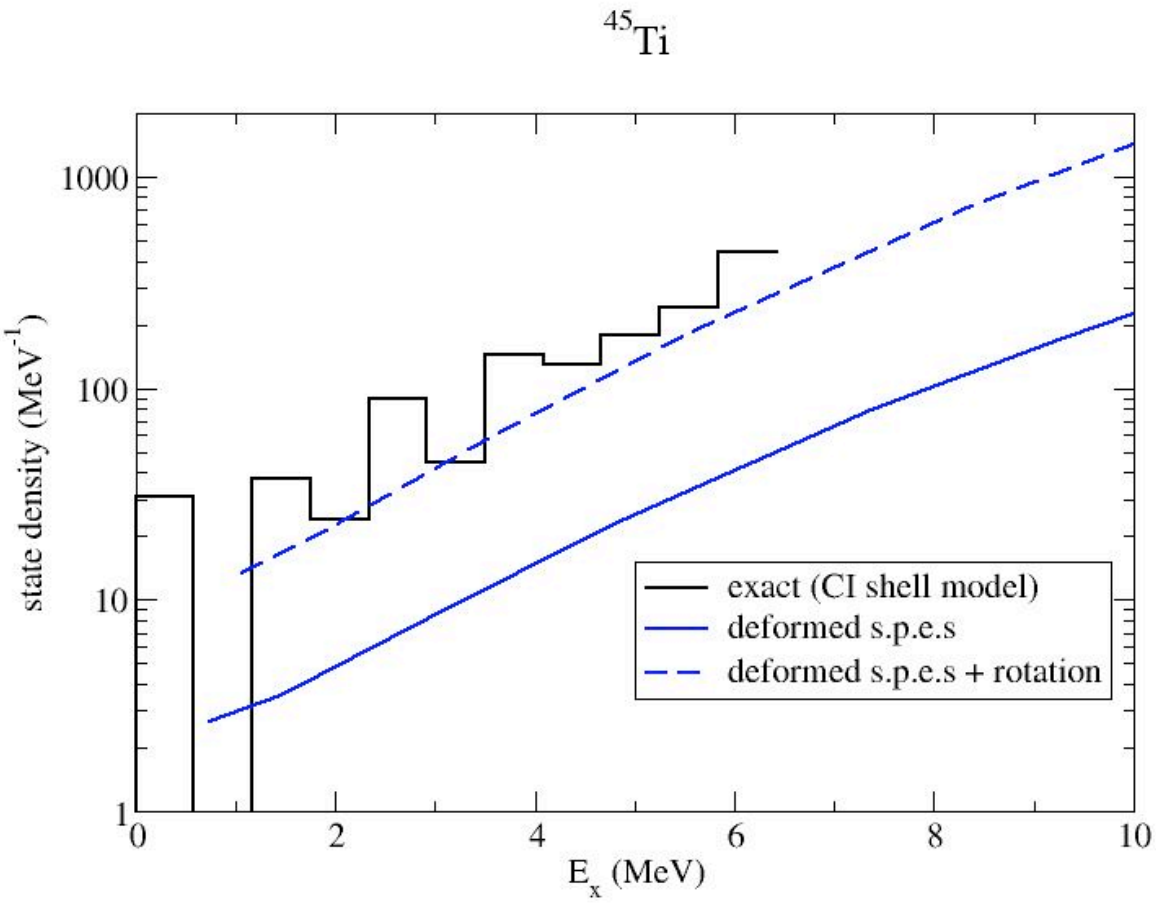
Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density



Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density



Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density

Method requires further investigation / validation:

Are other corrections (pairing/vibration) needed?
What about strong shape coexistence?
What about spin-cutoff factor?
What about spurious c.m. motion?

Nonetheless, a promising, cheap method to get level densities